UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Institut de Robòtica i Informàtica Industrial

iMOVE: Development of a hybrid control interface based on sEMG and movement signals for an assistive robotic manipulator

Marina Ramon Borràs

Master's Thesis

Directors:

Joan Lobo Prat

Adrià Colomé Figueras

Ponent: Carlos Ocampo Martinez

Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona

ETSEIB

Master's Degree in Automatic Control & Robotics

Barcelona, October 2020

Abstract

For many people with upper limb disabilities, simple activities of daily living such as drinking, opening a door, or pushing an elevator button require the assistance of a caregiver; which reduces the independence of the individual.

Assistive robotic systems controlled via human-robot interface could enable these people to perform this kind of tasks autonomously again and thereby increase their independence and quality of life. Moreover, this interface could encourage rehabilitation of motor functions because the individual would require to perform its remaining body movements and muscle activity to provide control signals.

This project aims at developing a novel hybrid control interface that combines remaining movements and muscle activity of the upper body to control position and impedance of a robotic manipulator.

This thesis presents a Cartesian position control system for KINOVA Gen3 robotic arm, which performs a proportional-derivative control low based to the Jacobian transpose method, that does not require inverse kinematics.

A second control is proposed to change the robot's rigidity in real-time based on measurements of muscle activity (sEMG). This control allows the user to modulate the robot's impedance while performing a task.

Moreover, it presents a body-machine interface that maps the motions of the upper body (head and shoulders) to the space of robot control signals. Its uses the principal component analysis algorithm for dimensionality reduction.

The results demonstrate that combining the three methods presented above, the user can control robot positions with head and shoulders movements, while also adapting the robot's impedance depending on its muscle activation.

In the future work the performance of this system is going to be tested in patients with severe movement impairments.

Acknowledgments

First of all, I would like to acknowledge Dr Joan Lobo and Dr Adrià Colomé for accepting me in their project, for giving me the opportunity to join IRI and for guiding me through all this process. This thesis wouldn't be possible without their global vision, priceless advice and experience.

From the UPC-ETSEIB, I would like to acknowledge Dr Carlos Ocampo for accepting to supervise this project, allowing me to realize the thesis at IRI.

Thanks also to all the people in the Perception and Manipulation Group: Sergi, Miguel, Alberto, David. For their advice, patience, support and knowledge, without you, this work wouldn't have been the same.

Finally, I would like to thank my beloved family and friends, for their unconditional support during this journey. Thank you for always being there.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

BMI Body-Machine Interface

COBOT Collaborative robot

DoF Degrees of Freedom

EMG Electromyography

IMU Inertial Measurement Unit

MSE Mean Squared Error

PCA Principal Component Analysis

PCs Principal Components

PRM People with Reduced Mobility

sEMG Surface Electromyography

Symbols

m	Cartesian space's degrees of freeedom
n	Robot's degrees of freeedom
$\boldsymbol{q}$	Robot's joints position
$\dot{\boldsymbol{q}}$	Robot's joints velocity
$\ddot{\boldsymbol{q}}$	Robot's joints acceleration
$\boldsymbol{\tau}$	Robot's joints torque
$\boldsymbol{B}$	Inertia matrix
$\boldsymbol{B_\chi}$	Inertia matrix in Cartesian space
$\boldsymbol{C}$	Centrifugal and Coriolis forces matrix

$\boldsymbol{F_v}$	Viscous friction coefficients diagonal matrix
$\boldsymbol{V_\chi}$	Velocity terms vector in Cartesian space
$\boldsymbol{g}$	Gravitational forces vector
$\boldsymbol{g_\chi}$	Gravitational forces vector in Cartesian space
$\boldsymbol{F}$	Force-torque vector acting on the robot's end-effector
$\boldsymbol{K_P}$	Matrix of proportional terms
$\boldsymbol{K_D}$	Matrix of derivative terms
$\boldsymbol{e}$	End-effector position and orientation error vector
$\dot{\boldsymbol{e}}$	End-effector linear and angular velocity error vector
$\boldsymbol{e_p}$	End-effector position error vector, in $\boldsymbol{e}$
$\boldsymbol{e_o}$	End-effector orientation error vector, in $\boldsymbol{e}$
$\boldsymbol{p}, \boldsymbol{p_d}$	End-effector, actual and desired, position vector
$\boldsymbol{v}, \boldsymbol{v_d}$	End-effector, actual and desired, velocity vector
$\boldsymbol{w}, \boldsymbol{w_d}$	End-effector, actual and desired, angular velocities vector
$\boldsymbol{R}, \boldsymbol{R_d}$	Rotation matrix, actual and desired
$\boldsymbol{m}$	Mass parameter of the impedance model
$\boldsymbol{b}$	Damping parameter of the impedance model
$\boldsymbol{k}$	Stiffness parameter of the impedance model
$\boldsymbol{F_e}$	Environment force acting on the robotic arm

1 | Introduction

This chapter introduces this master's thesis. Section 1.1 presents the motivation of the project. Then, Section 1.2 describes its specific objectives. Finally, a brief description of the thesis structure along with a summary of each of its chapters is found in Section 1.3.

1.1 Motivation

Robotic systems could potentially be very useful in helping people with severe physical disabilities. For instance, for many people with tetraplegia, simple activities required in daily living such as drinking, opening a door, or pushing an elevator button, require the assistance of a caregiver, which reduces independence of the individual. An assistive robotic system could enable people with severe movement disabilities to achieve greater independence and thereby increase quality of life.

Another challenge for people with motor impairments is the rehabilitation process, which aims at maintaining or recovering motor functions. To encourage the continued use of their motor functions, an individual's remaining body movements and muscle activity can be captured to provide control signals for the assistive device.

Therefore, the principal motivation of this project is to define a control system for an assistive robot to give, to people with severe physical disabilities, the opportunity to realize their daily tasks alone while recovering motor functions. To support this recovering the robotic system requires to employ the user remaining moves and its muscle activation as inputs to the control. Moreover, to apply the system in any situation without no danger, it must give to the user the ability to decide the robot's behaviour when there are physical interactions with the environment.

1.2 Objectives

In this project we propose to develop a novel approach that combines movement and sEMG[1] signals from the operator to control a robotic manipulator: the remaining upper-body

[1]The signal obtained of surface electrodes positioned on skin regions immediately above the muscle tissue that measure EMG (the electrical potential produced during muscle contractions).

motion are used to control the end-effector position and the sEMG signals are used to control multi-directional impedance. The specific objectives of this thesis are the following:

- Design of positionvelocity controller in Cartesian Space
- Implementation of an admittance control to adapt the robot's rigidity for different situations by sEMG signals
- Creation of a map between body movements and the robot's end-effector space

1.3 Thesis outline

This document is divided in the following sections:

- *Chapter 2: State Of The Art*

  This chapter explores the literature regarding the most common techniques used when implementing assistive manipulators. It covers examples of robotic arms, adaptive impedance control and body-machine interfaces.

- *Chapter 3: Theoretical background*

  This section is devoted to show some basic mathematical formulation used in other chapters of the thesis.

- *Chapter 4: Experimental Set-Up*

  This chapter presents the hardware and software employed in the project.

- *Chapter 5: Implementation*

  This chapter exposes the different implementations done during the project: starting with a proof of concept in one robot joint and ending at the final solution that achieves full-arm Cartesian position control with adaptive admittance.

- *Chapter 6: Economic study and Environmental footprint*

  This chapter manifests the analysis of the impact of the project both economically and environmentally.

- *Chapter 7: Conclusions*

  Finally, a discussion about the obtained results is presented.

2 | State Of The Art

This chapter introduces the state of the art of different techniques applied to assistive manipulators. It presents assistive robotic arms systems, some impedance control papers focused on adaptability and finally, studies on body movements which provide position control signals.

2.1 Assitive robotic arms

The number of assistive robots available on the market is limited. Examples of such assistive robotic devices are the Manus and the iARM (Figure 2.1a) by Exact Dynamics [1] and the JACO Arm (Figure 2.1b) by Kinova Robotics [2]. Such systems are generally controlled by a joystick, allowing the operator to move the arm in Cartesian space. A mechanical gripper, which can be opened and closed, is used to interact with objects. Although joysticks offer a rather intuitive interface, they are usually not suitable for individuals with severe movement impairments, and are limited to a stiff control of the end-point position.

(a) iARM

(b) JACO

Figure 2.1: iARM and JACO mounted on powered wheelchairs

2.2 Adaptive Impedance Control

Compliant human-robot interaction is an essential component for integrating robots into everyday life. Many daily activities require variable impedance, and humans desire the same functionality when interacting with robots [3]. Impedance control allows safe interaction with uncertain environments, enhancing both utility and viability [4].

Surface electromyography (sEMG) has been identified as a candidate for naturally controlling variable impedance [5] [6]. During co-contraction[1], muscle activation signals detected by sEMG correlate with the stiffness of the corresponding joints [7]. To date, very few studies have investigated multi-directional impedance with myoelectric signals [8] [9] [10].

On one hand, paper [8] proposes a novel Tele-Impedance algorithm that replicates the human's arm endpoint stiffness in robot. The following image exposes their set-up along with the scheme of the system, where it is observed that the stiffness control parameter is obtained directly from processing EMG signals.

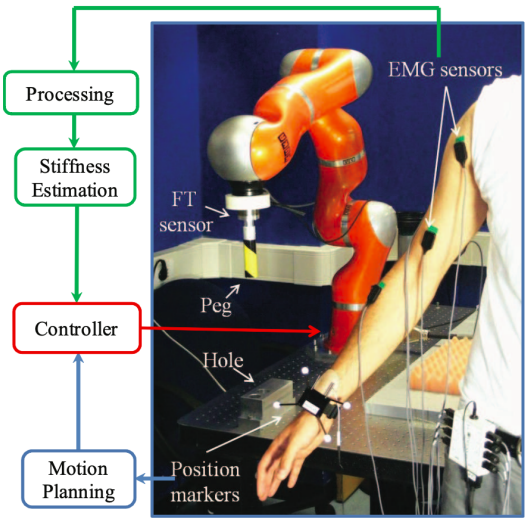


Figure 2.2: Peg-in-Hole experimental setup. KUKA light weight robotic arm, EMG electrodes, peg, hole, position tracking markers and F/T sensor are shown

On the other hand, paper [10] proposes a framework for multi-directional impedance control with sEMG to enhance compliant interactions. Next, example tasks from robot demonstrations are shown, where it is seen the arm behaviour depending on the subject stiffness.

[1]Simultaneous contraction of both the agonist and the antagonist muscle around a joint to hold a stable position.

(a) Anisotropic (b) Anisotropic (c) Isotropic (d) Isotropic

Figure 2.3: Example tasks from robot demonstration [10]. (a-b) Directional stiffness. (a) The subject provided high stiffness in the direction of the camera while moving to the right with low stiffness. (b) The arm was easily moved to the left due to the low stiffness. (c-d) Non-directional stiffness. The subject provided uniform stiffness to prevent the robot from moving in any direction.

2.3 Body-Machine Interface

Body-machine interfaces (BMI) use signals from body movements to control external devices. The majority of studies on BMI measure movement kinematics using IMUs to control external devices such as computer robot arms [11] [12], screen cursors [13] [14] [15], drones [16] and powered wheelchairs [17]. The mapping between body motions and the motion of the external device is usually performed using PCA[2].

The figure below shows a complete schematic of BMI [13]:

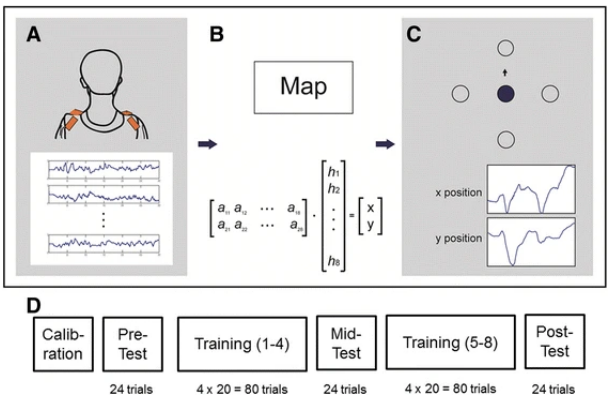


Figure 2.4: Schematic of the body-machine interface [13]. **(A)** 4 IMUs are attached to the upper body that measure movement of the upper body. **(B)** These signals are transformed by a map A into the x-y position of a cursor. **(C)** The position of the cursor is displayed on a screen and the participant is asked to move the cursor to different targets. **(D)** Schematic of the experimental protocol in one session – after an initial calibration phase, participants are asked to perform a series of training and test blocks to evaluate motor learning. The calibration block was only performed on the first experimental session, whereas the remaining blocks were performed in each experimental session.

[2]The principal component analysis is the process of computing the data's principal components and using them to perform a change of basis.

In [14] there is presented the following study set up:

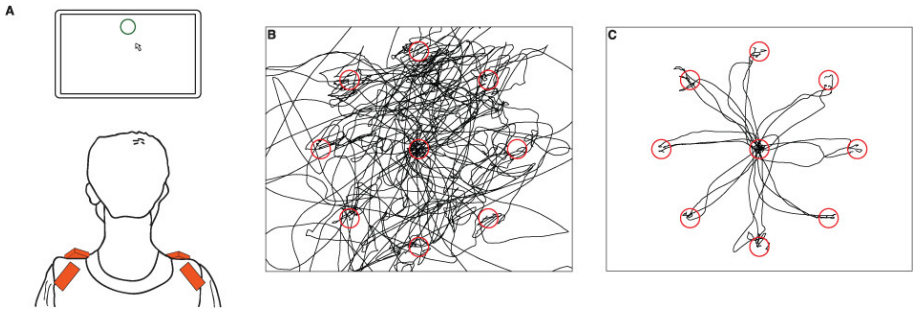
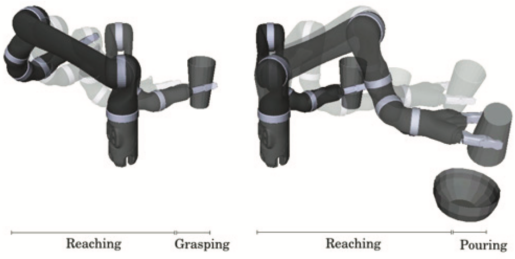
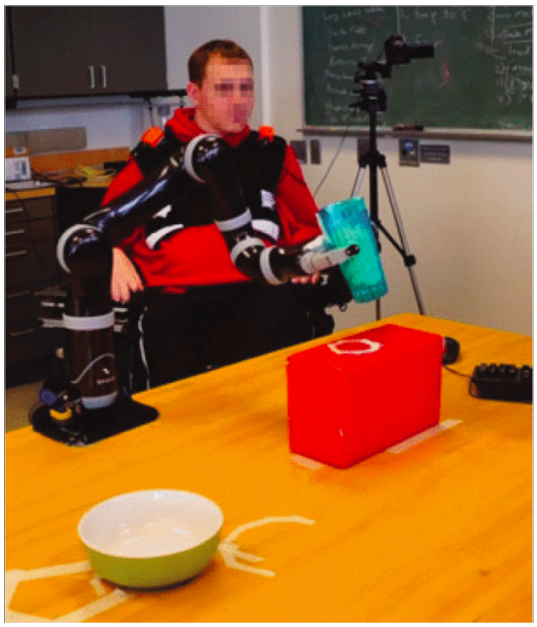


Figure 2.5: Study set-up [14]. **(A)** Participants sat in front of a computer monitor wearing 4 inertial measurement units on the shoulders. Signals from the sensors were mapped into the 2D position of a computer cursor used to perform the different tasks; Sample reaching movement from one participant **(B)** first session versus **(C)** last session. The circles illustrate the center and peripheral targets, and each line represents the cursor's path during one center-out reach.

As an example of a real application, next figure shows a user with Spinal Cord Injury (SCI) that is controlling JACO arm for a manipulation task (shown in Figure 2.6a) [11]:



(a) Illustration of the piecewise segments associated with the experimental task.



(b) An SCI user controlling the robot with the BMI during the experimental task.

Figure 2.6: Example of a real manipulation task application using BMI.[11]

3 | Theoretical background

This thesis chapter presents the theoretical framework used in the different experiments. It introduces, first, the concept of impedance and admittance control; second, a full-arm control method in Cartesian space and finally, the principal component analysis algorithm.

3.1 Impedance and Admittance control

Impedance control is an approach that imposes desired dynamic behaviour when there is a physical interaction between robot end-effector and environment. The inverse of the impedance is admittance. It is the ratio of position to force.

The main difference between them is that impedance controls the force after motion or deviation from a set point is measured and in contrast, admittance, imposes position, so it defines the motions that result from a force input.

The system impedance is modelled as a spring-mass-damper with stiffness $\boldsymbol{k}$, mass $\boldsymbol{m}$ and damping $\boldsymbol{b}$; connecting the actual position ($\boldsymbol{x_a}$) of the robotic arm to the desired position ($\boldsymbol{x_d}$) [3]. The equation of motion for the system is given by:

$$\boldsymbol{m\ddot{x}_a = -b\dot{x}_a - k(x_a - x_d) + F_e} \tag{3.1}$$

where $\boldsymbol{F_e}$ represents the environment force acting on the robotic arm. This equation defines the motion of the system under the influence of external forces.

Considering the above equation (3.1), the following figure presents the control scheme for impedance and admittance:

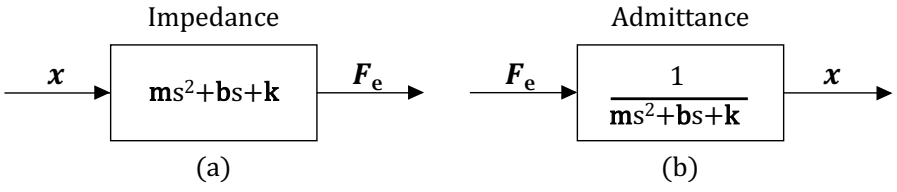


Figure 3.1: Impedance (a) and Admittance (b) control schemes.

One application in modern-day prosthetic limbs is proposed by Amy A. Blank et al. [3], where there is investigated that user-selectable prosthesis impedance properties could improve the user's ability to interact effectively with a variety of environments. Next Figure 3.2 shows how the prosthesis impedance is modelled as a spring-mass-damper.

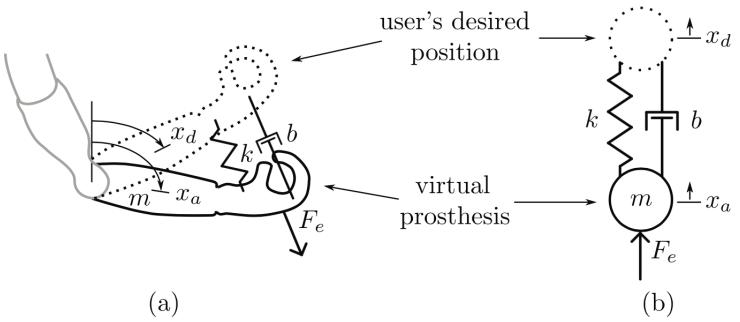


Figure 3.2: Model of virtual prosthesis impedance [3]. (a) Schematic of an idealized one-degree-of-freedom prosthetic arm. We represent the virtual prosthesis as a point mass m at position x_a. The prosthesis impedance is modeled as a spring with stiffness k and a damper with damping b connecting the actual position to the desired position x_d. F_e represents environment force applied on the prosthesis. (b) For these studies, the virtual prosthesis is represented in Cartesian space.

3.1.1 Robot's rigidity concept

In robotic arms, controlling the robot's physical interaction with the environment leads to the concept of a robot's rigidity. This one refers to the level of stiffness applied to interact with the environment, to behaves:

- Rigid: so, to remain at a given position, whatever the external forces exerted on the robot.
- Compliant, which allows deviations from its equilibrium position, depending on the applied external force.

The applications that employ adjustable compliance achieve to interact effectively with a variety of environments.

3.2 Jacobian transpose

The robotic arm's Jacobian transpose relates the robot's end-effector Cartesian space directly with its joint space. In robots control, to apply this relation means to obtain a unique robot configuration and so, to resolve the *inverse kinematics problem.*

In terms of dynamics, in [18] it is observed that having computed the total kinetic and potential energy of a system taking the derivatives required by Lagrange equations and considering the absence of external forces; the *joint space dynamic model* can be represented by the following equation of motion:

$$\boldsymbol{B(q)\ddot{q} + C(q,\dot{q})\dot{q} + F_v\dot{q} + g(q) = \tau} \tag{3.2}$$

where $\boldsymbol{q, \dot{q}, \ddot{q}} \in \mathbb{R}^n$ are joints position, velocity and acceleration; $\boldsymbol{B(q)} \in \mathbb{R}^{nxn}$ is the inertia matrix which is symmetric and positive definite; $\boldsymbol{C(q,\dot{q})} \in \mathbb{R}^{nxn}$ accounting for centrifugal and Coriolis forces; $\boldsymbol{F_v} \in \mathbb{R}^{nxn}$ denotes the diagonal matrix of viscous friction coefficients; $\boldsymbol{g(q)} \in \mathbb{R}^n$ is the vector of gravitational forces and $\boldsymbol{\tau} \in \mathbb{R}^n$ expressing the joint torques that are applied to the robot, being n it's degrees of freedom (DoF).

Due that the above expression matrices are functions only of manipulator position it can be considered as the *configuration-space equation.* Although, as explained in Chapters 10 and 11 from [19], it might be desirable to express the dynamics of a manipulator with respect to Cartesian variables in the following general form:

$$\boldsymbol{B_\chi(q)\ddot{\chi} + V_\chi(q,\dot{q}) + g_\chi(q) = \mathcal{F}} \tag{3.3}$$

where $\boldsymbol{\mathcal{F}} \in \mathbb{R}^m$ is a force-torque vector acting on the end-effector of the robot, and $\boldsymbol{\chi} \in \mathbb{R}^m$ is an appropriate Cartesian vector representing position and orientation of the end-effector. Analogous to the joint space quantities, $\boldsymbol{B_\chi(q)} \in \mathbb{R}^{mxm}$ is the Cartesian inertia matrix, $\boldsymbol{V_\chi(q,\dot{q})} \in \mathbb{R}^m$ is the velocity terms vector (so, $\boldsymbol{(C+F_v)\cdot\dot{q}}$) in Cartesian space, and $\boldsymbol{g_\chi(q)}$ is a vector of gravity terms in Cartesian space.

Taking into consideration the above both expressions, and having in mind that the definition of the Jacobian is:

$$\boldsymbol{\dot{\chi} = J(q)\dot{q}} \tag{3.4}$$

where $\boldsymbol{\dot{\chi}}$ is the end-effector velocity. It can be noted that the fictitious forces acting on the end-effector, $\boldsymbol{\mathcal{F}}$, could in fact be applied by the actuators at the joints, $\boldsymbol{\tau}$, by using the relationship:

$$\boldsymbol{\tau = J^T(q)\mathcal{F}} \tag{3.5}$$

with this (3.5) result it is proved that the Jacobian transpose ($\boldsymbol{J^T}$) maps Cartesian forces acting at the hand into equivalent joint torques.

3.2.1 End-effector forces calculation

The Jacobian transpose is presented as a method for designing a trajectory-following control system, that can cause the manipulator to follow the desired position trajectory. So, it is necessary to ensure an error tending to zero between the desired and actual robot's position. It is sufficient to define a proportional–derivative (PD) control law with

the position and velocity vectors as feedback. Therefore, end-effector forces obtention from (3.5) is as follows:

$$\boldsymbol{\mathcal{F}} = \boldsymbol{K_D e} + \boldsymbol{K_P \dot{e}} \tag{3.6}$$

where $\boldsymbol{K_P}$ and $\boldsymbol{K_D} \in \mathbb{R}^{mxm}$ are diagonal matrices all non-negative, that denote the coefficients for the proportional and derivative terms respectively and $\boldsymbol{e}$, $\boldsymbol{\dot{e}} \in \mathbb{R}^m$ represent end-effector position and velocity error.

Position error: $\boldsymbol{e}$

Computing the position error of (3.6) isn't as direct as it would be in the joint space by simple subtraction, in Cartesian control, it is necessary to differentiate two terms: error in pose and error in orientation. The first one is calculated as:

$$\boldsymbol{e_p} = \boldsymbol{p_d} - \boldsymbol{p} = \begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} - \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{3.7}$$

where $\boldsymbol{p_d}$, $\boldsymbol{p}$ denote desired and actual robot pose.

On the other hand, the error in orientation can be computed in different manners depending on its representation: Euler angles, axis–angle or quaternion. One simple option is by using the rotation matrix, which can be derived for each case.

Considering $\boldsymbol{R} = \begin{bmatrix} \boldsymbol{n} & \boldsymbol{s} & \boldsymbol{a} \end{bmatrix}$ and $\boldsymbol{R_d} = \begin{bmatrix} \boldsymbol{n_d} & \boldsymbol{s_d} & \boldsymbol{a_d} \end{bmatrix}$ like the actual and desired rotation matrices, the error in orientation is calculated as follows:

$$\boldsymbol{e_o} = \frac{1}{2}(\boldsymbol{n} \times \boldsymbol{n_d} + \boldsymbol{s} \times \boldsymbol{s_d} + \boldsymbol{a} \times \boldsymbol{a_d}) \tag{3.8}$$

where $\frac{1}{2}$ term represents the usually used [20] relation between meters and radians (1 m = 2 rad), added because pose and orientation have different units.

Finally, taking (3.7) and (3.8), position error can be expressed as $\boldsymbol{e} = \begin{bmatrix} \boldsymbol{e_p} \\ \boldsymbol{e_o} \end{bmatrix}$.

Velocity error: $\boldsymbol{\dot{e}}$

The end-effector linear and angular velocity error is obtained directly by the subtraction of desired and actual velocity in Cartesian space:

$$\boldsymbol{\dot{e}} = \boldsymbol{v_d} - \boldsymbol{v} = \begin{bmatrix} \boldsymbol{\dot{p}_d} \\ \boldsymbol{w_d} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\dot{p}} \\ \boldsymbol{w} \end{bmatrix} \tag{3.9}$$

where $\boldsymbol{v_d}$, $\boldsymbol{v} \in \mathbb{R}^m$ denote desired and actual linear and angular velocities; $\boldsymbol{\dot{p}_d}$, $\boldsymbol{\dot{p}}$ $\in \mathbb{R}^m$ represent desired and actual linear velocities and $\boldsymbol{w_d}$, $\boldsymbol{w} \in \mathbb{R}^m$ are the desired and

actual angular velocities.

Applying the relation mentioned in (3.4), end-effector velocity can be obtained directly by joint velocities and the geometric Jacobian.

3.3 Principal Component Analysis (PCA)

Principal component analysis is a mathematical algorithm commonly used for dimensionality reduction (Figure 3.3) by projecting each data point onto a new set of variables to obtain lower-dimensional data while retaining most of the variation in the data set. It accomplishes data set reduction by identifying directions, called principal components, along which the variation in the data is maximal.

Before extracting the principal components (PCs), the original data is normalized (by subtracting the mean) so that each variable contributes equally to the analysis.

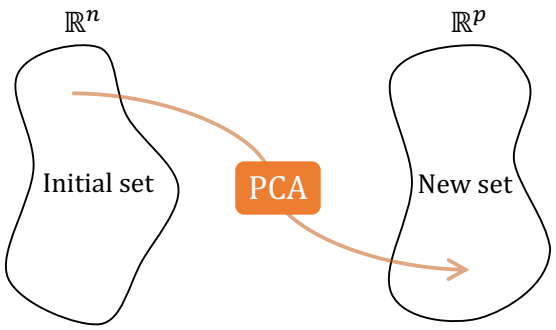


Figure 3.3: PCA dimensionality reduction, with n>p.

Then, as there are as many principal components as there are variables (n) in the data, PCs are ordered by variance: The first component, PC 1, represents the direction of the highest variance of the data. The direction of the second component, PC 2, represents the highest of the remaining variance orthogonal to the first component. This can be naturally extended to obtain the required number of components (p) which together span a component space covering the desired amount of variance.

Figure 3.4 presents an example of PCs obtention, where the original data set has 3 dimensions and applying PCA it is reduced to a 2 dimensional space:

Since principal components describe specific directions in the data space, each one depends on certain amounts on each of the initial data; so every PC is a linear combination of all original variables.

PCA algorithm aims to use the *Feature vector* that represents these linear combinations,

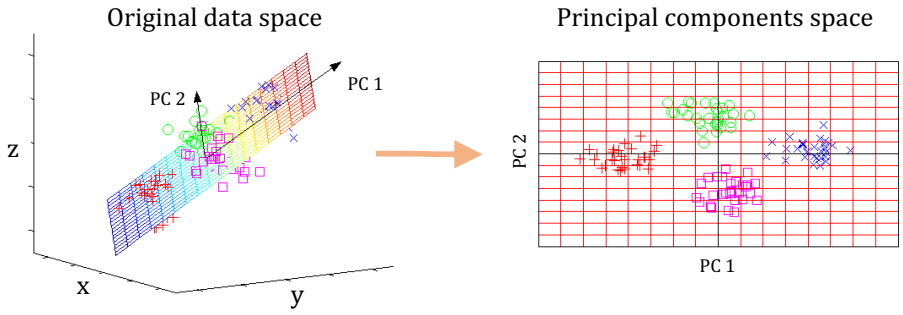


Figure 3.4: Example of dimensionality reduction using PCA: from a 3D data set to 2D.

to reorient the data from the original axes to the ones represented by the principal components, as seen next:

$$FinalDataSet = FeatureVector^T \cdot NormalizedOriginalDataSet^T \tag{3.10}$$

4 | Experimental Set-Up

This chapter presents the hardware and software employed in the project implementation. It gives an overview of each device and program, along with its main characteristics.

4.1 Hardware

The devices manipulated during this project include a robotic arm, where we test all the controllers, and two sensors: Xsens MVN Awinda and Myo Armband, used as input signals to the hybrid control interface.

4.1.1 KINOVA Gen3 Ultra lightweight

The robotic arm chosen for this project is the KINOVA Gen3 Ultra lightweight with 7 degrees of freedom (DoF). It is the third generation of Kinova, a company specialized, in particular, in assistive robots.

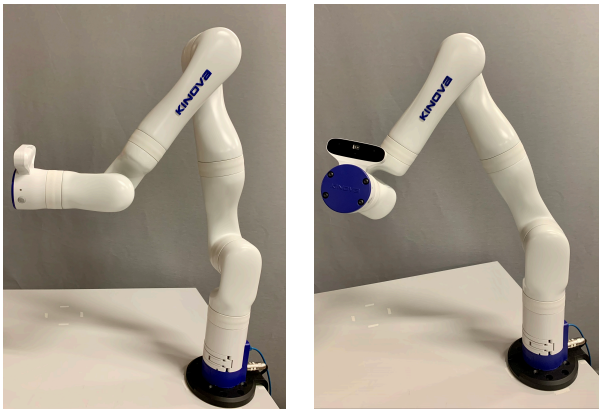
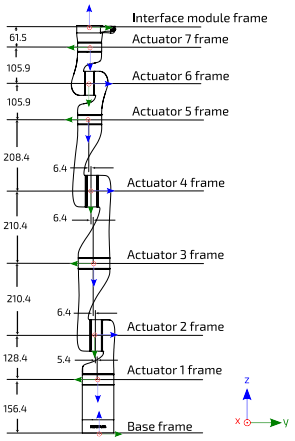


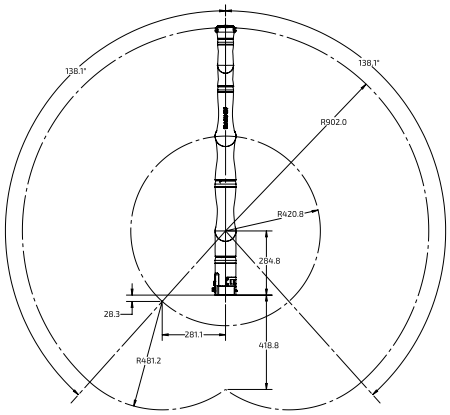
Figure 4.1: KINOVA Gen3 Ultra lightweight with 7 DoF.

KINOVA Gen3 is designed for safety, efficiency and control in real-world environments. Its general characteristics involve being ultra-lightweight (8.2 kg), portable, power-efficient (36 W) and having the best payload-to-weight ratio.

Figure 4.2a provides a schematic diagram of the robot and its physical dimensions. On the right, Figure 4.2b shows KINOVA Gen3 workspace; the set of all locations in the three-dimensional space reachable by the end-effector (all measurements in mm):



(a) 7 DoF spherical wrist frames dimensions



(b) 7 DoF robot nominal workspace

Figure 4.2: KINOVA Gen3 schematic diagram and its nominal workspace.

Regarding its software: the robot is enabled by KINOVA KORTEX, the Kinova software framework and application development platform. Its APIs are currently provided for the following languages: C++, Python and MATLAB. Moreover, there are ROS packages covering most of the same functionalities.

In terms of robot control, KINOVA Gen3 count with two levels:

- **High-level (40 Hz)**, which control is simpler to use, and offers added protection. Its connection with the base is a TCP session for most API calls, it offers a long list of control modes: Cartesian control is one of them.
- **Low-level (1000 Hz)**, which offers lighter and faster commands and finer-grained control. Its connection with the base is a UDP session for BaseCyclic calls, which demands a command for each actuator in the robot. So it does not count with Cartesian control mode, just joint control.

4.1.2 Xsens MVN Awinda

The Xsens MVN inertial motion capture system is an easy to use, cost efficient system for full-body human motion capture. It is a completely portable; it is not restricted to a studio or lab. It can be used anywhere: outside, in the office, and on the work floor.

The model of Xsens MVN available in the laboratory is the Awinda (Figure 4.3), a full-body inertial kinematic measurement system incorporating 17 IMUs or MTw

(wireless motion trackers)) with 3D accelerometers, gyroscopes, magnetometers (3D compass), and a barometer (pressure sensor).

Figure 4.3: Xsens MVN Motion Capture model Awinda.

The MVN system is controlled by a software application called MVN Analyze/Animate, a 64-bit application for Windows 10 which can be used for real-time viewing and recording. It provides to the user all segments information as well as joint angle data, the centre of mass and factory calibrated sensor data. Figure 4.4 shows an example of how the system data is processed and represented.

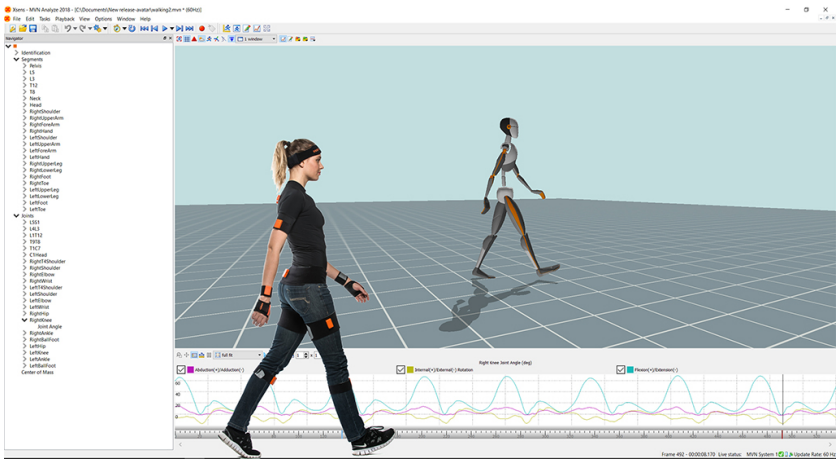


Figure 4.4: Representation of Xsens motion capture in MVN Animate.

This thesis implements the sensor to give reference positions to control. First, we use directly the right elbow angle and second, we apply a map between 3 IMUs orientations and a 3-D Cartesian space data.

4.1.3 Myo Armband

Myo armband is a wearable device by Thalmic Labs Inc. provided with eight electromyographic (EMG) electrodes, a 9-axes inertial measurement unit (IMU) and a transmission module. It sends the data related to the detected signals, via Bluetooth Low Energy technology.

This device represents a complete electronic platform that detects in real-time the main signals related to forearm activity (muscles activation and forearm movements in the three-dimensional space) and sends these data to the connected devices.

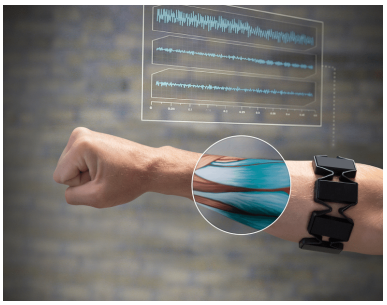


Figure 4.5: Myo Armband detecting forearm activity in real-time.

This thesis implements the sensor to control the adaptive admittance of KINOVA by EMG signals: Myo armband records EMG signals with a limited bandwidth of about 200 Hz.

4.2 Software

This project employs the Robotic System Toolbox of MATLAB to test the controller in a simulation environment. It provides tools and algorithms for simulating and testing KINOVA Gen3 (Figure 4.6), which allows to quickly study the designed control before being applied to the real robot.

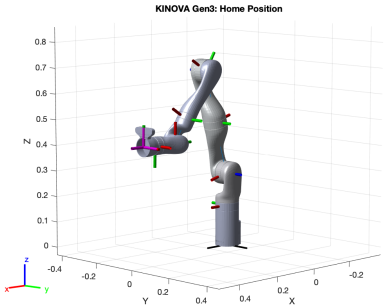


Figure 4.6: MATLAB: KINOVA Gen3 simulation using the Robotic System Toolbox.

The programming language chosen to control the robotic arm is C++, applied in both High-level and Low-level control.

5 | Implementation

This chapter presents the different implementations done during the thesis to achieve the final solution. The first one is a proof of concept on a robot joint. Next, Cartesian position control is designed and tested in simulation and then, in the real environment. The third implementation presents a Body-Machine Interface (BMI) using Xsens MVN's data. Finally, the complete solution which proposes a full-arm Cartesian position control with adaptive admittance is exposed.

5.1 Control of a joint with adaptive admittance

This project aims at developing a method that combines body movements and sEMG signals from the operator to control a robot. This section reveals the first approach performed at low scale as a proof of concept.

Here the objective is to control the position and admittance of just one joint of KINOVA Gen3 (shown in 4.1.1). That means to program joint positions and moreover, to change the robot's rigidity in real-time.

Methodology

KINOVA is set to low-level to command actuator positions at a frequency of 1000 Hz. Joint 6 (See Figure 5.18) is the one controlled, chosen due to its similarity with a human elbow.

The reference position is given by the Xsens MVN motion capture system (presented in 4.1.2). This experiment employs the right elbow's angle to determine the desired position of actuator number 6. As the joint hasn't the complete elbow movement, the input position is limited to avoid robot collisions. (See an application example in Figure 5.4)

Robot's rigidity can be changed by applying an admittance control. The idea of adopting different stiffness for each situation requires the use of an external sensor to control its level. This project proposes an adaptive admittance control which allows adapting the robot's rigidity respect to muscle activation. These sEMG signals are

captured by the Myo armband (section 4.1.3).

Considering the above methodology, the next Figure 5.1 shows the block diagram proposed to control the position of joint number 6 with adaptive admittance:

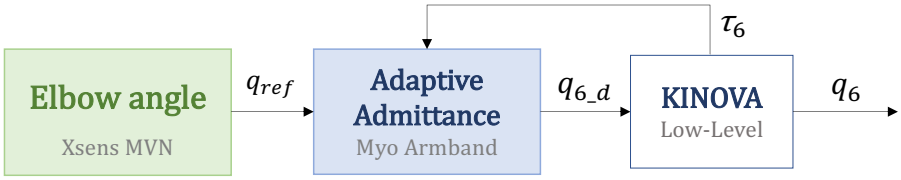


Figure 5.1: Block scheme: Control of a joint position with adaptive admittance.

The adaptive admittance block employs a method exposed in the theoretical background (section 3.1) adjusted to the thesis purpose. Its detailed control scheme is shown in the following figure:

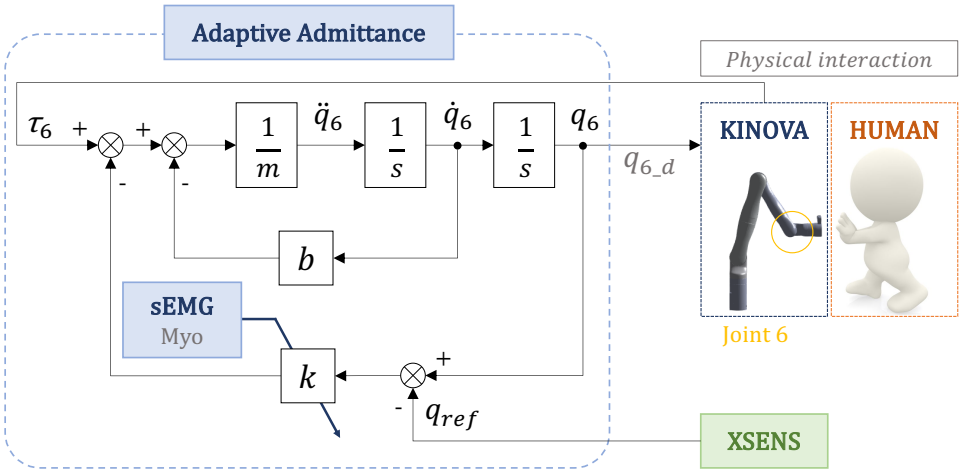


Figure 5.2: Block scheme: Adaptive admittance control in joint space.

The diagram shows that admittance control imposes value of position resulting from an input force. In this approach, position $\boldsymbol{q_{6\_d}}$ is determined by the environment force $\boldsymbol{\tau_e}$ acting on the robot's actuator number 6.

As seen in Figure 5.18, KINOVA's actuator admittance is modelled as a spring-mass-damper. So, it connects the actuator position ($\boldsymbol{q}$) with the position reference ($\boldsymbol{q_{ref}}$) given by the Xsens MVN with a determined stiffness.

The Myo is employed to change the robot stiffness ($\boldsymbol{k}$). Each EMG signal is filtered by the root mean square with a window size of 100 points. To compute the map between the sum of the 8 sEMG and the value of $\boldsymbol{k}$, the maximum value that the user can perform

is obtained to normalize the data. The following equation shows the function used to obtain stiffness values in the range $[\boldsymbol{k_{min}}\ \boldsymbol{k_{max}}]$:

$$k = k_{min} + \frac{|EMG_{sum}|}{|EMG_{max}|}(k_{max} - k_{min}) = k_{min} + EMG_{norm}(k_{max} - k_{min}) \quad (5.1)$$

This control achieves that KINOVA's joint 6 rigidity adapts to muscle activation.

Results

First, to analyze the designed control, a set of positions given by the right elbow of the Xsens MVN are tracked. In each one, the spring-mass-damper values are tested to determine the best performance possible without varying robot's rigidity. The following figure shows the results:

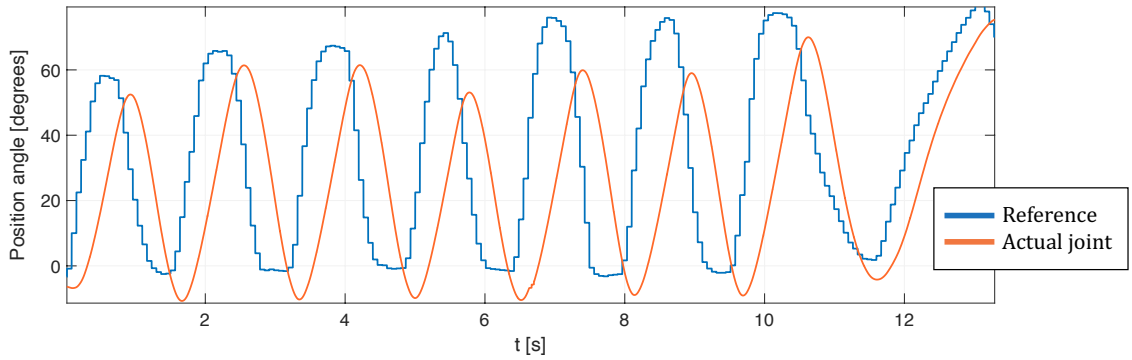


Figure 5.3: Tracking right elbow angles with fixed admittance.

It is observed that the admittance control system designed allows the robot joint to follow the desired right elbow angles with a maximum error of 14 degrees if the velocity of tracking is high. Next table presents the selected spring-mass-damper parameters for modelling the KINOVA's actuator number 6:

Parameter	**Value**	**Units**
m	0.05	kg
b	0.3	Ns/m
k_{min}	3	N/m
k_{max}	15	N/m

Table 5.1: First approach: Actuator 6 mass-spring-damper parameters.

The experimental set-up is presented below: Xsens MVN motion capture plus the Myo armband in the right forearm; along with two examples of its implementation. They

show how the position of the joint number 6 mimics precisely the right elbow angle when the level of stiffness is high:

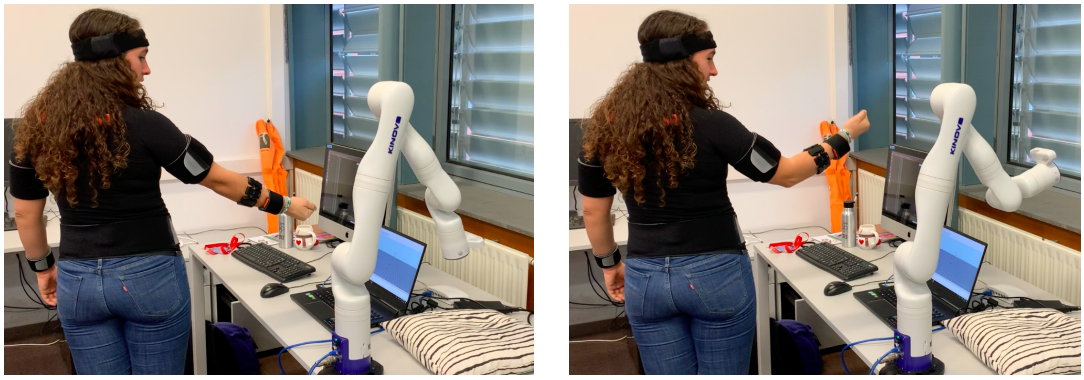


Figure 5.4: KINOVA's joint position control applied to mimic the right elbow angle.

Second, to analyze the control joint performance with changes in its rigidity, new angles tracking is implemented with adaptive admittance. It is achieved by varying the stiffness k in relation to sEMG signals obtained from muscle activity in the right forearm.

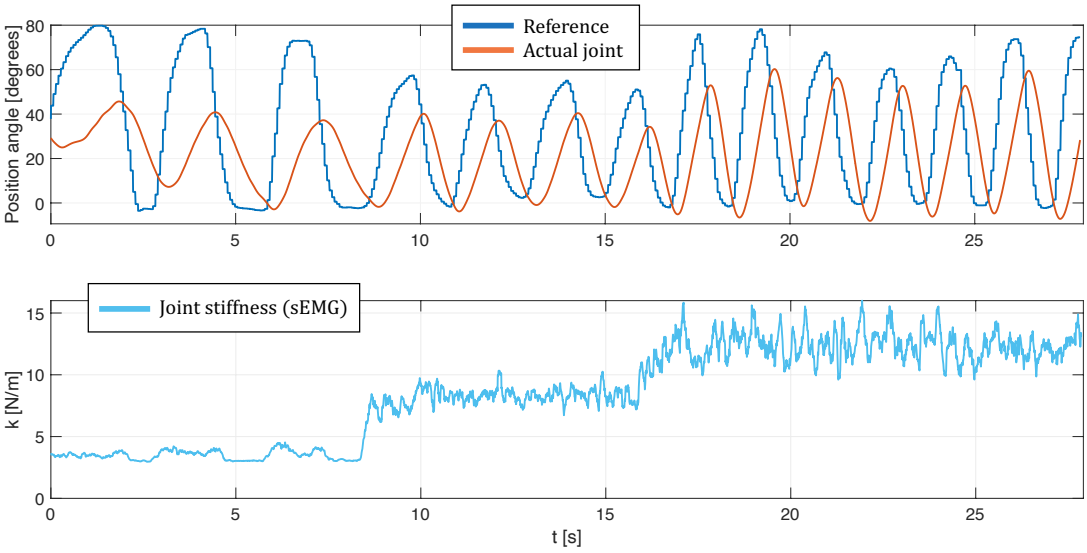


Figure 5.5: Tracking right elbow angles with adaptive admittance.

It is seen that now the robot stiffness, that change tracking precision, adapts to the force that the user wants to apply in the action. When the gain of k is low an error of almost 40 degrees is observed. That behaviour, although it can be seen incoherent, is very useful in assistive robots due to the ability to move it with high compliance.

Finally, it is done an experiment to test the approach while an external force acts on

the joint, results displayed in Figure 5.6. The first 9 seconds no external force is applied to the joint so, the torque (in green) has a constant range of values. During this part, the joint's ability to follow precisely the position just depends on k. The second part consists of showing how the joint behaviour changes when the robot is affected by external forces during its trajectory. The grey lines separate diverse performances with an object placed at around 20 degrees. It can be seen that depending on the robot stiffness different results on the joint's position control are achieved:

1. With low stiffness the joint is not able to follow the trajectory, it stops until the desired position is below the object.

2. Incrementing the stiffness but also the force that the object exerts, the situation is the same as before.

3. When the torque obtained is lower than 6 Nm and the k is higher than 10, the joint acts with rigidity and moves the object to follow the trajectory.

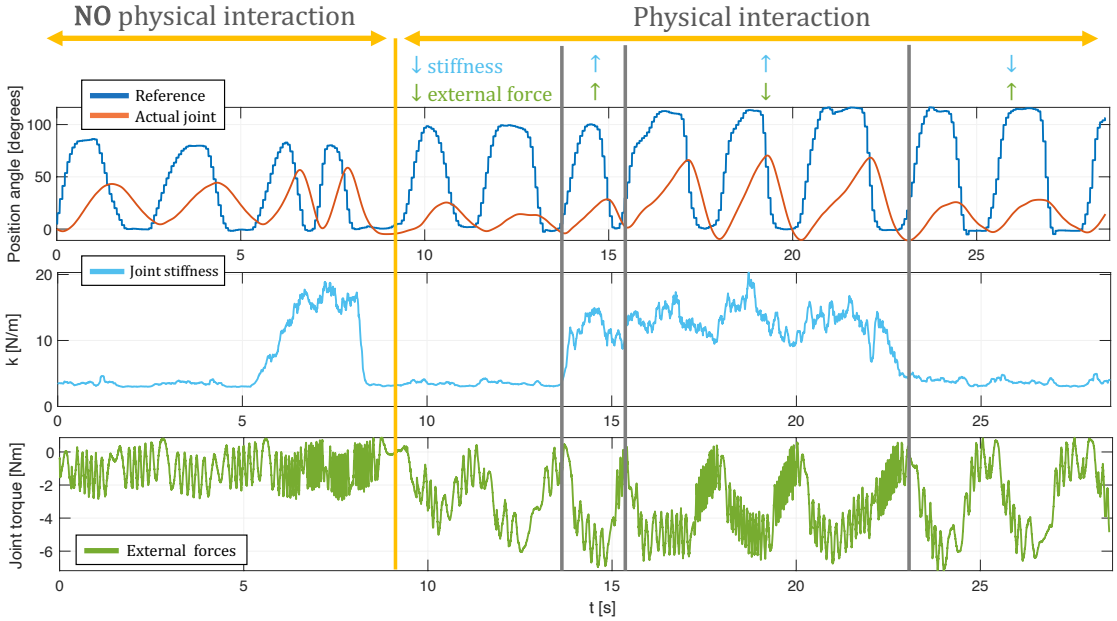


Figure 5.6: Tracking right elbow angles with adaptive admittance while external forces interacting.

With this last implementation, it can be concluded that the control of a joint with adaptive admittance works as expected. It is able to track desired angle positions while adapting its rigidity. So, in case of detecting external forces, the robot does not continue following the trajectory if the force exerted by the operator is not high enough.

5.2 Low-level Cartesian position control

In robotics, a manipulator is usually controlled by signals of joint space position $\boldsymbol{q} \in \mathbb{R}^{\boldsymbol{n}}$, being $\boldsymbol{n}$ the number of robot's DoF (Degrees of Freedom). Although, when controlling specific movements, being able to give the desired end-effector orientation and position in Cartesian space is very useful.

This section presents the implementation of a Cartesian position control applied to KINOVA Gen3 at low-level servoing mode (computing frequency of 1000 Hz).

Methodology

To implement a Cartesian position control means to determine desired robot positions in Cartesian variables: pose (x, y, z) and orientation (for example Euler angles: roll, pitch, yaw).

KINOVA's low-level control API demands a command for each actuator in the robot. It allows setting joints position, velocity or torque.

Performing this control in KINOVA requires to apply the desired positions (Cartesian space) as a robot's configurations (joint space). So, this implementation needs to relate the robot's end-effector Cartesian space with its joint space.

In Section 3.2, we have explained a methodology that maps Cartesian forces acting to the hand ($\boldsymbol{\mathcal{F}}$) into equivalent joint torques ($\boldsymbol{\tau}$). The figure below shows graphically this relationship applied to KINOVA:

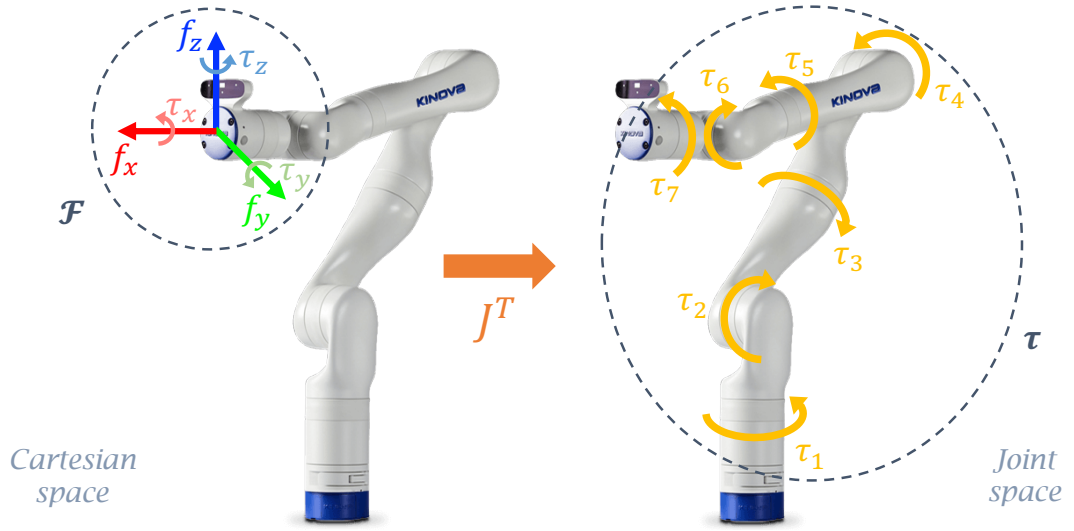


Figure 5.7: Graphical representation of Jacobian transpose relationship (Equation 3.5).

The Cartesian position control uses the Jacobian transpose ($\boldsymbol{J^T}$) to program the robot in torque configurations ($\boldsymbol{\tau_d}$). Controlling a robot by torque commands implies to consider its gravitational compensation. The following figure shows a detailed scheme of this control's implementation:

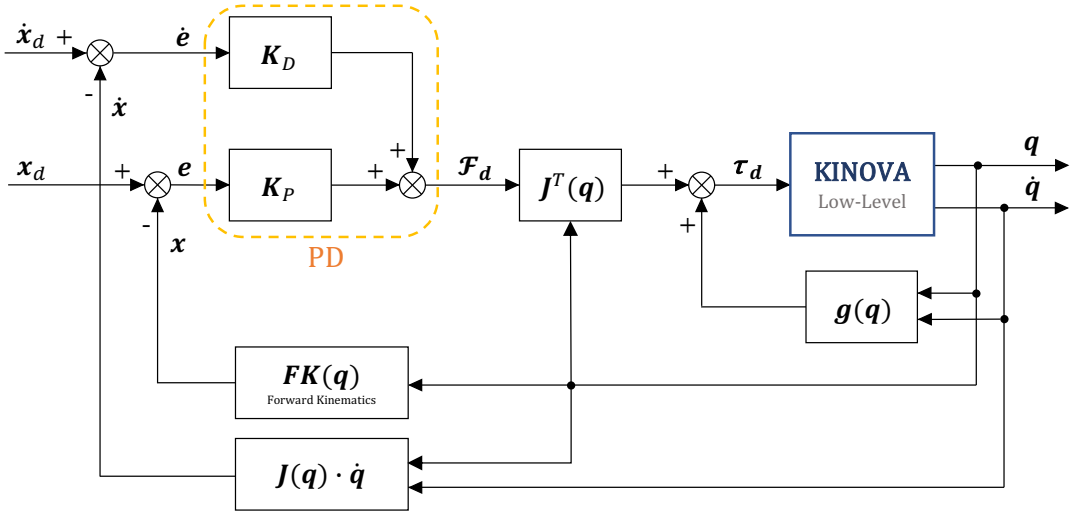


Figure 5.8: Block scheme: Cartesian position control employing the Jacobian transpose.

As observed in the scheme, Forward Kinematics and geometric Jacobian have been computed to obtain, respectively, KINOVA's position ($\boldsymbol{x}$) and velocity ($\boldsymbol{\dot{x}}$) in the Cartesian space.

This implementation applies a proportional-derivative (PD) controller. It takes the end-effector position and velocity errors ($\boldsymbol{e}$,$\boldsymbol{\dot{e}}$) and obtains the end-effector forces ($\boldsymbol{\mathcal{F}_d}$) needed to follow the position reference ($\boldsymbol{x_d}$). $\boldsymbol{K_P}$ and $\boldsymbol{K_D} \in \mathbb{R}^{6x6}$ are diagonal matrices, that denote the coefficients for the proportional and derivative terms respectively. The $\boldsymbol{K_P}$ gains are tuning parameters and the ones of $\boldsymbol{K_D}$ are computed from them:

$$\boldsymbol{K_D} = 2\sqrt{\boldsymbol{K_P}} \tag{5.2}$$

5.2.1 Simulation: MATLAB

The proposed Cartesian position control is applied first in a simulated environment to test its quality before performing it on the actual robot. The KINOVA Gen3's kinematics and dynamics are implemented with the MATLAB Robotic System Toolbox [21].

Results

In order to apply the control scheme seen in Figure 5.8 and achieving the best performance, the PD parameters have to be defined. To ensure precise position and orientation control,

a K_p gains study has been performed. It consists of computing the mean squared error (MSE) of a trajectory several times with different gains. The resulting graphics are presented in the next figure:

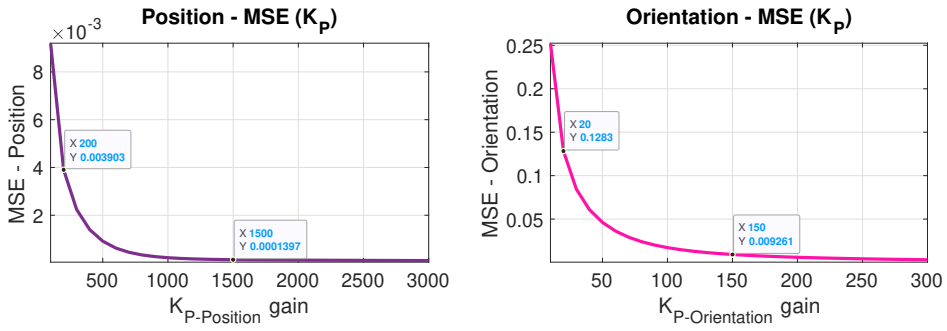


Figure 5.9: Simulation: Comparison of the mean squared error (MSE) obtained while tracking same positions with different $\boldsymbol{K_P}$ gains.

To analyze this control a set of positions has been tracked twice varying its gains. From the above graphics, the chosen high gains are: $\boldsymbol{K_P}$ = diag([1500 1500 1500 150 150 150]) and the low ones: $\boldsymbol{K_P}$ = diag([200 200 200 20 20 20]). Next figure shows the results:

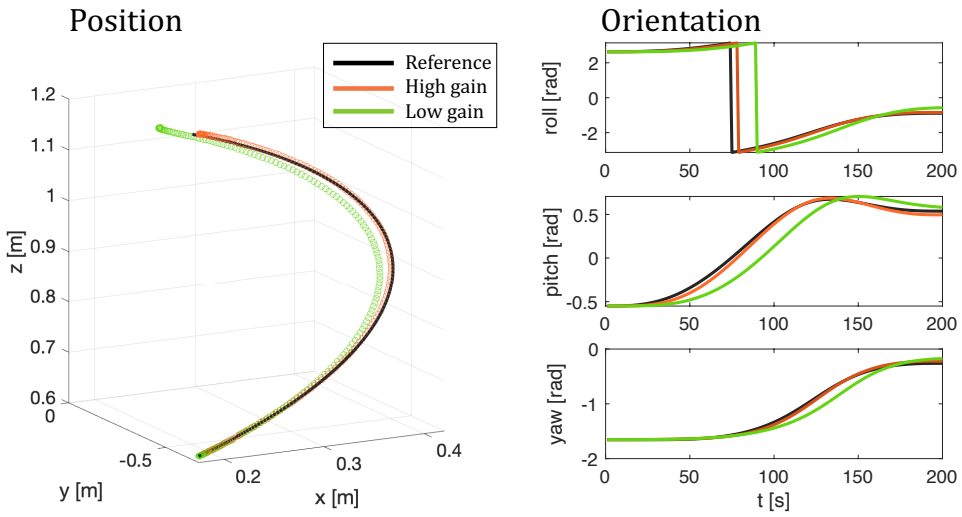


Figure 5.10: Simulation: Tracking positions with the Cartesian position control designed.

It is observed that the control system designed seems to work for tracking a trajectory. It verifies that employing high gains, the resulting position and orientation errors are small.

5.2.2 Real environment: C++

Here the aim is to implement the control law, already tested in simulation, in the real robot; to analyze the KINOVA's ability to track Cartesian positions in real-time.

Pinocchio library [22]) is used to compute torque compensations: considering the gravity and also, Coriolis and centrifugal effects on the robot's dynamics. Regarding forward kinematics and jacobian, they are computed using the transformation matrices provided in KINOVA's User Guide [23].

Results

Following the same strategy as in simulation: to analyze the implementation in the real environment, a trajectory has been tracked, in this case just once with the higher gains to obtain the most precise results. This trajectory is generated as a set of Cartesian positions that determine a circumference's perimeter with free orientation to not conflict with position tracking.

- **Reference - Position:**
  Circumference in the XY plane of radius 20 cm, centred in KINOVA's home position $\begin{bmatrix}0.547 & 0.003 & 0.431\end{bmatrix}$ m.

In the next figure, the controller proportional and derivative terms that adjust the robot's rigidity in the real environment are the same as in the simulation:

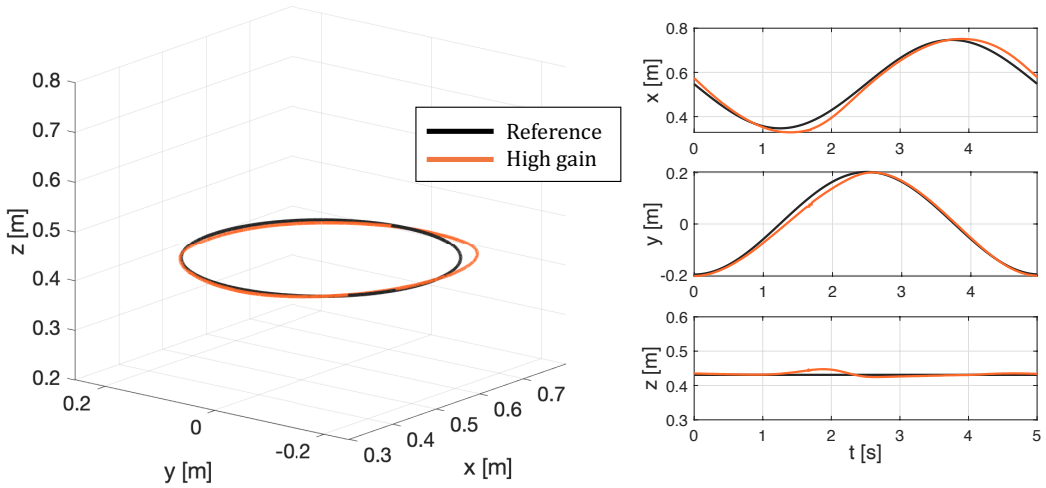


Figure 5.11: Real environment: Tracking a circumference with the controller designed.

It is observed that, effectively, the control system designed allows the robot to follow the desired Cartesian positions. It is confirmed that with high gains, the error obtained is lower than 4 cm.

5.3 Body-Machine Interface (BMI)

The project aims to design a system that enables people with severe movement disabilities to control a robotic manipulator. This section proposes a body-machine interface (BMI) that takes upper-body motions and represents them in Cartesian space. This method allows determining KINOVA's end-effector positions with body movements acquired by the Xsens MVN.

Methodology

The BMI principle is to use a dimensionality reduction technique to map high dimensional measurements of body movement to the state of a low dimensional controller. In this application, the dimensionality reduction is done by a Principal Component Analysis (PCA) explained in section 3.3.

Experimental Set-up. The user sits in front of KINOVA and has to guide its end-effector to specific targets. To detect upper-body moves: this implementation uses 3 IMUs (inertial measurement units) to measure shoulders and head movements. For each IMU, signals of orientation (roll, pitch, and yaw angles) are acquired. The IMUs used in this study are Xsens MTw (Wireless Motion Tracker) sensors.

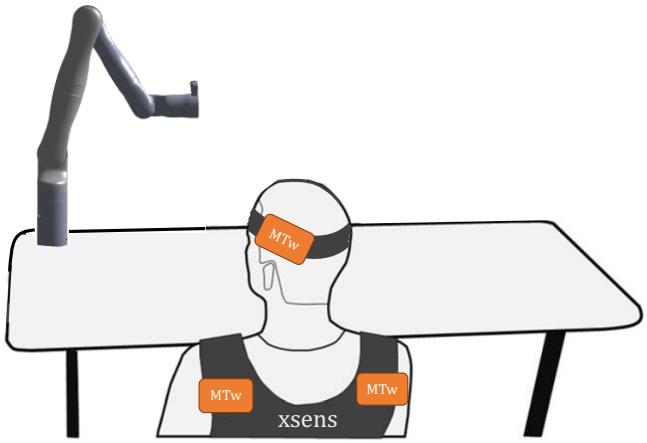


Figure 5.12: Schematic of the experiment. The user wears 3 IMUs (inertial measurement units): 1 in the head and 1 in each shoulder, to control the end-effector of KINOVA Gen3.

Controlling the robot arm using IMU signals. This approach maps the 9-IMU angles into the 3-D robot end-effector velocity as follows:

$$\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \boldsymbol{A} \cdot \begin{bmatrix} s1 & s2 & ... & s9 \end{bmatrix}^T \tag{5.3}$$

where $\boldsymbol{v_x}$, $\boldsymbol{v_y}$, $\boldsymbol{v_z}$ are the linear velocities [m/s] of Cartesian space, the matrix $\boldsymbol{A}$ is the *map* that transforms IMU signals into end-effector velocities, and s1..s9 are the 9 angles from the IMUs (3 sensors x 3 angles).

In the final solution, KINOVA is programmed with the Cartesian position control exposed in the last section. The control law is designed to follow position references, so, BMI outputs must be positions. The velocity signals obtained from equation 5.3 are integrated to obtain the Cartesian position applying the next equation:

$$\boldsymbol{x} = \boldsymbol{x} + \boldsymbol{v} \cdot dt \longrightarrow \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}_{PCA} \cdot dt \tag{5.4}$$

where $\boldsymbol{x}, \boldsymbol{v}$ denote position and lineal velocity in Cartesian space.

Protocol. *Calibration.* To determine the matrix $\boldsymbol{A}$ that maps orientation signals to Cartesian space, the user must perform a sequence of arbitrary body motions in which make exploratory movements with his upper body while maintaining a comfortable range of motion.

Next, the principal components analysis (PCA) is applied to extract the first 3 principal components (PCs) from the calibration data. No filtering or pre-processing is applied to the IMU signals. These three PCs formed the rows of the mapping matrix $\boldsymbol{A}$ (the x, y and z-component are controlled by motion along PC1, PC2 and PC3, respectively). The PCs are normalized, subtracting their respective mean, and multiplied by a factor chosen to provide reasonable velocities of the end-effector.

Moreover, IMUs can capture even small unintentional deviations (such as those due to breathing) from the resting posture and potentially affect the velocities. Therefore, a *dead-zone* is designed, so that robot starts moving only when the velocity exceeded this threshold.

Testing. This block aims to analyze the range and quality of BMI outputs in a simulation environment before commanding the real robot. The test is run in a ROS (Robot Operating System) node that publishes marker positions in rviz and consists of performing a centre-out reaching task with a screen cursor. The goal is to move the marker that simulates the robot (colored in blue) into the specified target (displayed on the screen with a green marker) as quickly as possible. (See an example on Figure 5.14)

Results

The following table shows the information about the principal components obtained during the calibration process of one user. It is important to remark that the calibration will be different for each user, so this mapping is not valid for every operator.

PCA	PC1	PC2	PC3	Total
Explained variance ratio	0.512	0.326	0.060	89.813 %
Singular values	3065.01	2443.51	1051.63	-

Table 5.2: Body-machine interface (BMI) parameters tuning.

The head and shoulders orientations obtained during the calibration process represent a 9-dimensional input to the PCA. Next figure presents in a 3-dimensional set the 89.81% of the original data total variance:

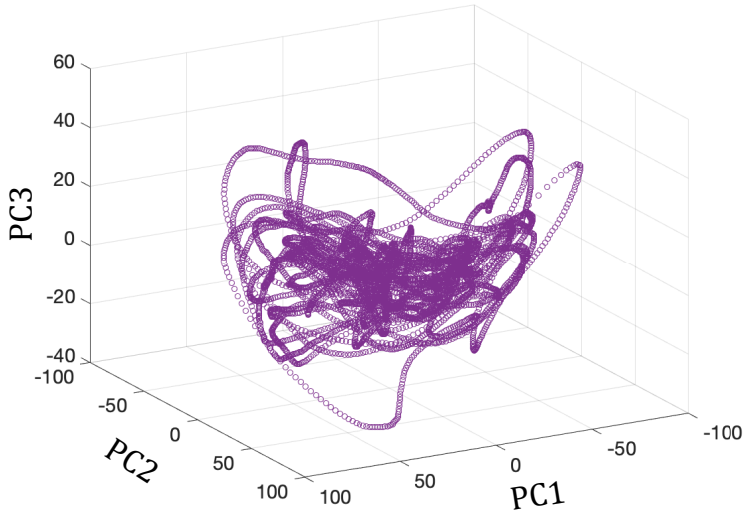


Figure 5.13: Visualization of the original orientations (in 9-D) in the new 3-dimensional space. It represents 89.81% of the original data total variance.

To provide consistent velocity commands to the robot end-effector, as commented in methodology, a dead-zone and gain are determined for each principal component. Considering the range of the principal components in Figure 5.13, the next table presents the determined parameters:

	v_x	v_y	v_z	Units
Principal component gain	0.33	0.33	1.0	m/s
Dead-zone	3.33	3.33	5.0	m/s

Table 5.3: Body-machine interface (BMI) parameters tuning.

The following figure shows the real experimental set-up for testing, where the operator is performing a centre-out reaching task with a screen cursor:

Figure 5.14: Testing: Real experimental set-up. The user wears the Xsens MVN Awinda, that have inertial measurement units in the head and shoulders , to control the blue marker (actual position) to reach the green one (goal).

To analyze the proposed body-machine interface, we present an example of a reaching task. Figure 5.15 shows the trajectory in 3-D determined by body-movements to achieve the goal point:

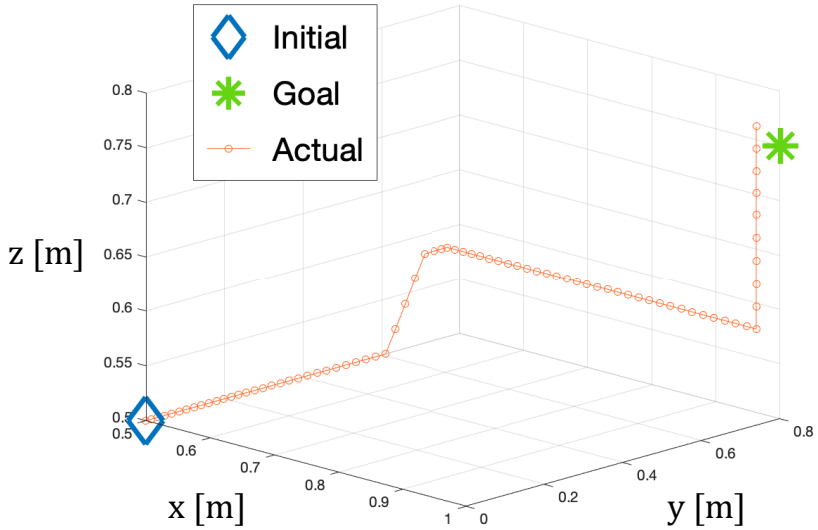


Figure 5.15: Testing: Example of the trajectory computed during one of the reaching tasks.

To understand better how the above trajectory is achieved by head and shoulders moves, we present the BMI inputs and outputs of the example. The following figure shows the set of body-movements (in Euler angles) and their corresponding Cartesian velocities:

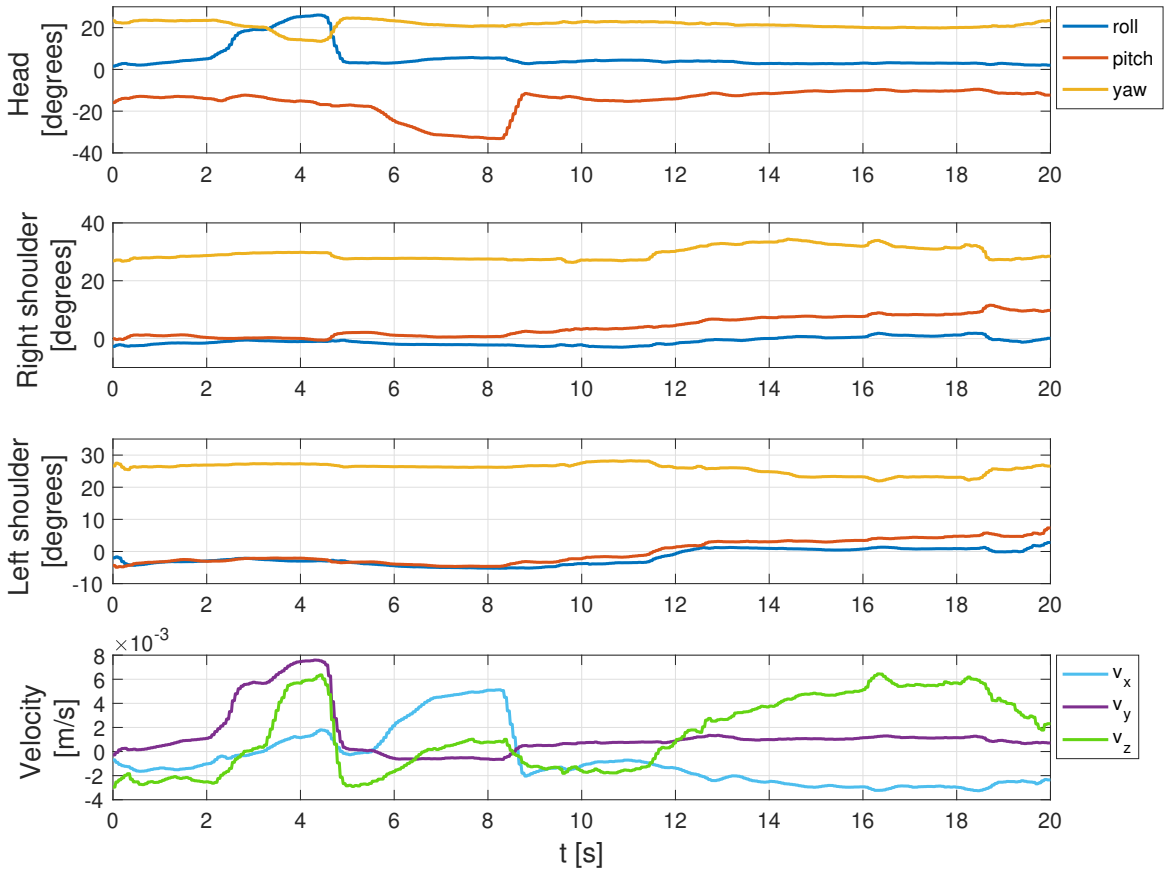


Figure 5.16: Testing: Body-machine interface inputs and outputs.

It is observed that the range of velocities obtained from the BMI is $[-0.4, 0.8]$ [cm/s], which are inside the Cartesian translation speed that KINOVA Gen3 can compute. To sum up, the designed body-machine interface can be used to control the real robot end-effector positions.

5.4 Final solution

This last section presents the project's final solution, which proposes a low-level Cartesian position control with adaptive admittance. It allows the operator to control the robot position with upper-body movements and to decide the robot's rigidity in each instant of time.

This application is truly useful for assistive robots because it lets the operator adapt the robot behaviour depending on the situation. Consequently, if an action requires high precision and there is no danger of collision, the operator can decide to move the robot with rigidity and accuracy. In contrast, when the task does not require precision, or it is done near people, the operator can control the robot to be compliant so, it stops when colliding and does not result dangerous.

Methodology

It implements all the methods presented in the last three sections in one approach: It combines the low-level Cartesian position control with the admittance control used in the proof of concept. Moreover, the reference positions are determined in real-time using the body-machine interface.

Next, to understand better the complete solution, there is a bit explanation of each method: its application in this approach, the hardware that employs and the data it works with:

- **Low-level Cartesian position control:**
  To apply a position control with inputs in end-effector (Cartesian) space and outputs in a 7 DoF joint space.
  $\longrightarrow$ *KINOVA Gen3* at low-level mode: data in joint space (frequency = 1000 Hz)

- **Admittance control:**
  To adapt the robot's rigidity by muscle activation. This block gives the inputs to Cartesian control.
  $\longrightarrow$ *Myo Armband*: sEMG signals

- **Body-machine interface (BMI):**
  To determine desired positions with upper-body movements.
  $\longrightarrow$ *Xsens MVN Awinda*: 3 IMUs (1 in the head and 2 in the shoulders)

This overview reveals that the Cartesian position control and the BMI are applied without any modification. In contrast, the admittance control in this approach requires to be computed in Cartesian space. The following figure shows a detailed scheme of the final solution implementation:

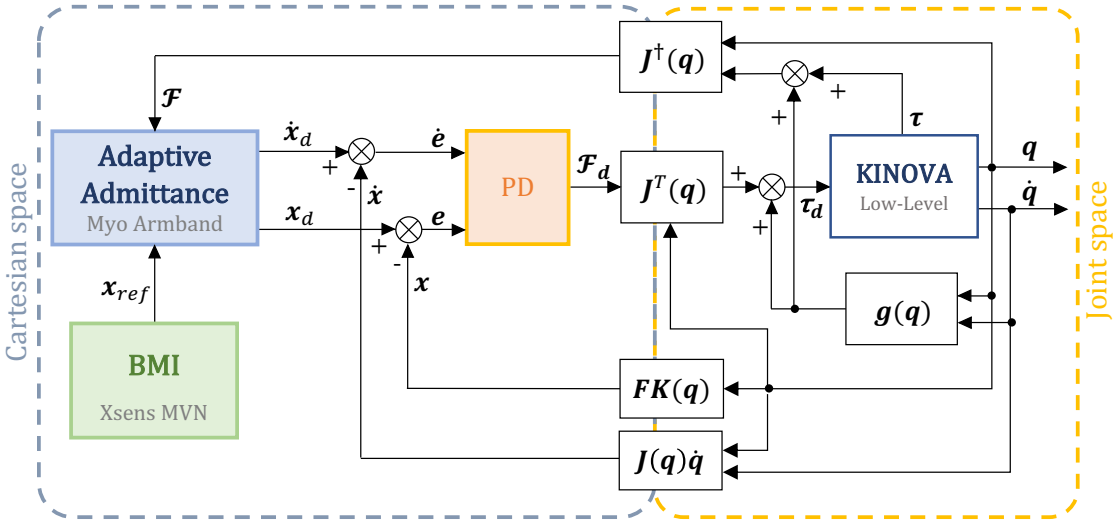


Figure 5.17: Block scheme: Cartesian position control by body-movements with adaptive admittance.

where $\boldsymbol{x}, \dot{\boldsymbol{x}} \in \mathbb{R}^6$ denote position and velocity in Cartesian space; $\boldsymbol{e}$,$\dot{\boldsymbol{e}} \in \mathbb{R}^6$ denote end-effector position and velocity errors; $\boldsymbol{\mathcal{F}} \in \mathbb{R}^6$ represents the end-effector forces (wrench and torques); $\boldsymbol{J} \in \mathbb{R}^{6x7}$ is the Jacobian matrix; $\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\tau} \in \mathbb{R}^7$ denote respectively actuators position, velocity and torque in joint space.

It is observed that the block of adaptive admittance takes the end-effector forces $\boldsymbol{\mathcal{F}}$ as input. So, to obtain it, the pseudoinverse of Jacobian ($\boldsymbol{J}^{\dagger}$) is computed and then, multiplied by the compensated joint torques of the robot. A detailed admittance control scheme in Cartesian space is shown in the following figure:

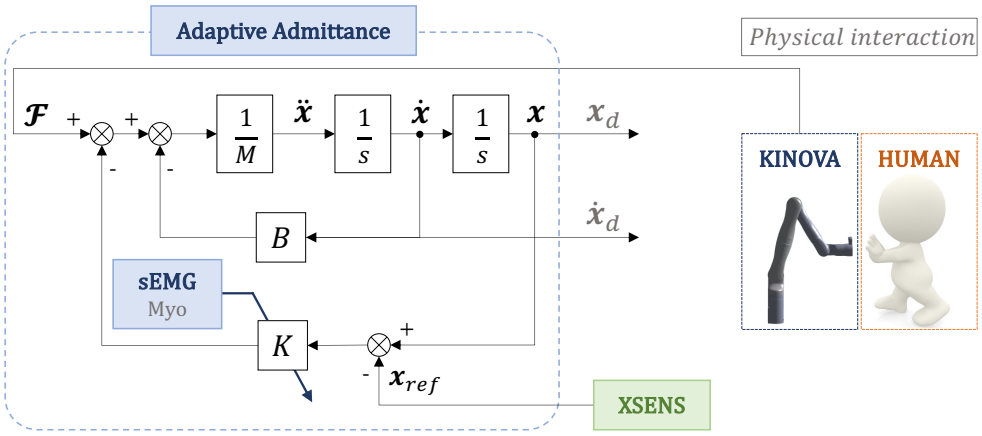


Figure 5.18: Block scheme: Adaptive admittance control in Cartesian space.

Results

The following figure presents the final experimental set-up, where the operator is controlling the robot positions with head and shoulders movements along while adapting its rigidity by sEMG signals:

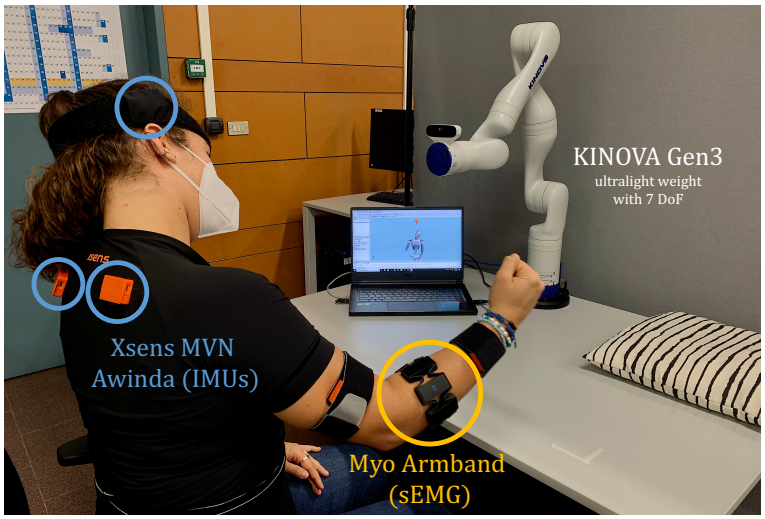


Figure 5.19: Final solution experimental set-up. The user wears the Xsens MVN Awinda, that have IMUs in the head and shoulders and the Myo Armband in the right forearm. It is performing head and shoulder movements and sEMG signals by muscle activation to control robot position and admittance.

Next table presents the parameters determined to achieve the best final solution's performance. It shows the proportional and derivative gains of the Cartesian position control; the admittance control parameters: spring-mass-damper, to model the KINOVA admittance:

PD Control	**Parameter**	**Value**	**Unit**
	$\boldsymbol{K_{Dposition}}$	1500	–
	$\boldsymbol{K_{Dorientation}}$	150	–
Admittance Control			
	$\boldsymbol{M}$	2	kg
	$\boldsymbol{B}$	20	Ns/m
	$\boldsymbol{K_{min}}$	100	N/m
	$\boldsymbol{K_{max}}$	600	N/m

Table 5.4: Final solution parameters tuning.

The BMI gains and thresholds that provide reasonable end-effector positions are shown in Table 5.3. These parameters have been determined in a simulation environment by performing several body-machine interface tests.

To analyze the hybrid control interface, several head and shoulders free movements with changes in right forearm muscle activation are realized. This performance allows observing the differences in the position's error obtained when the robot is programmed to behave compliant or rigid.

The following figure demonstrates that when the sEMG signals are small, so the robot behaves compliant, the error between the reference position and the actual KINOVA's position is significant. In contrast, when the robot stiffness is higher, the error becomes small, so the control interface can track more precisely the desired position.

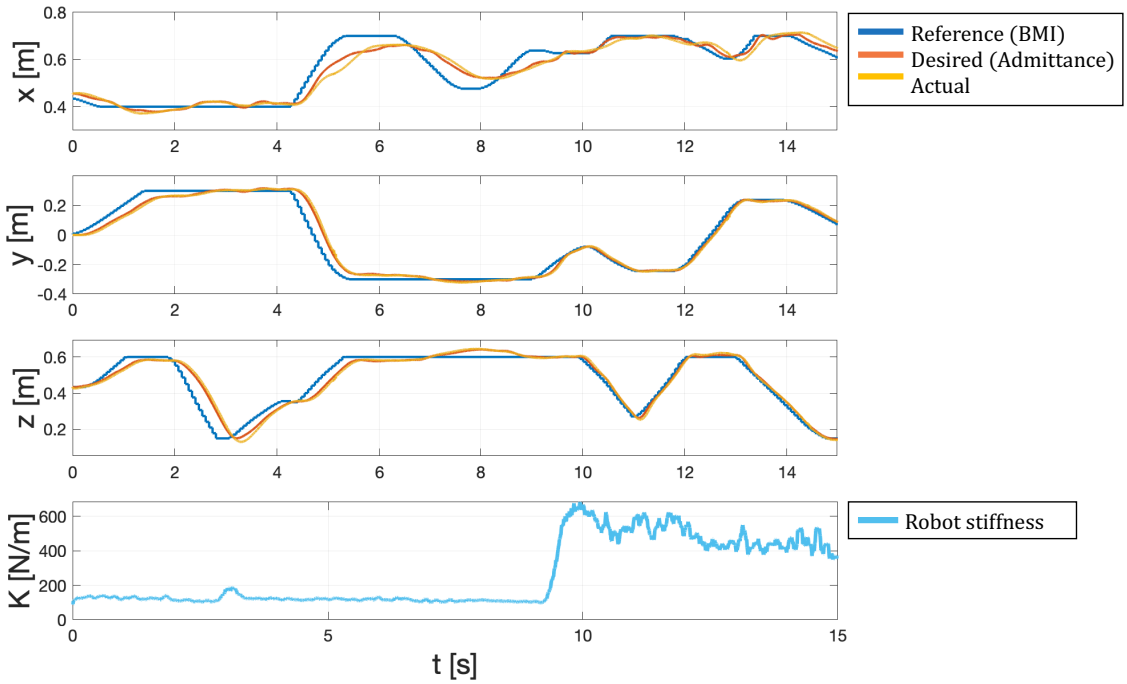


Figure 5.20: Cartesian positions tracking with adaptive admittance. Reference positions obtained applying a body-machine interface to head and shoulders movements. Desired positions determined by the robot's modelled system that takes BMI position as a reference and the sEMG as control signal.

It is seen that now the robot stiffness, that change tracking precision, adapts to the force that the user wants to apply in the action. When the gain of $\boldsymbol{k}$ is low an error of almost 40 degrees is observed. That behaviour, although it can be seen incoherent, is very useful in assistive robots due to the ability to move it with high compliance.

Finally, it is done an experiment to test the approach while an external force acts on the operational Y axis of the robot, results displayed in Figure 5.21. Until the first second no external force is applied to the robot so, the wrench detected (in green) is almost 0. During this part, the robot's ability to follow precisely the position just depends on $\boldsymbol{K}$. The second part consists of showing how the robot behaviour changes when it is affected by an external force in the y-axis during its trajectory. The grey lines separate diverse performances with an object placed at around y = 0.05 m.

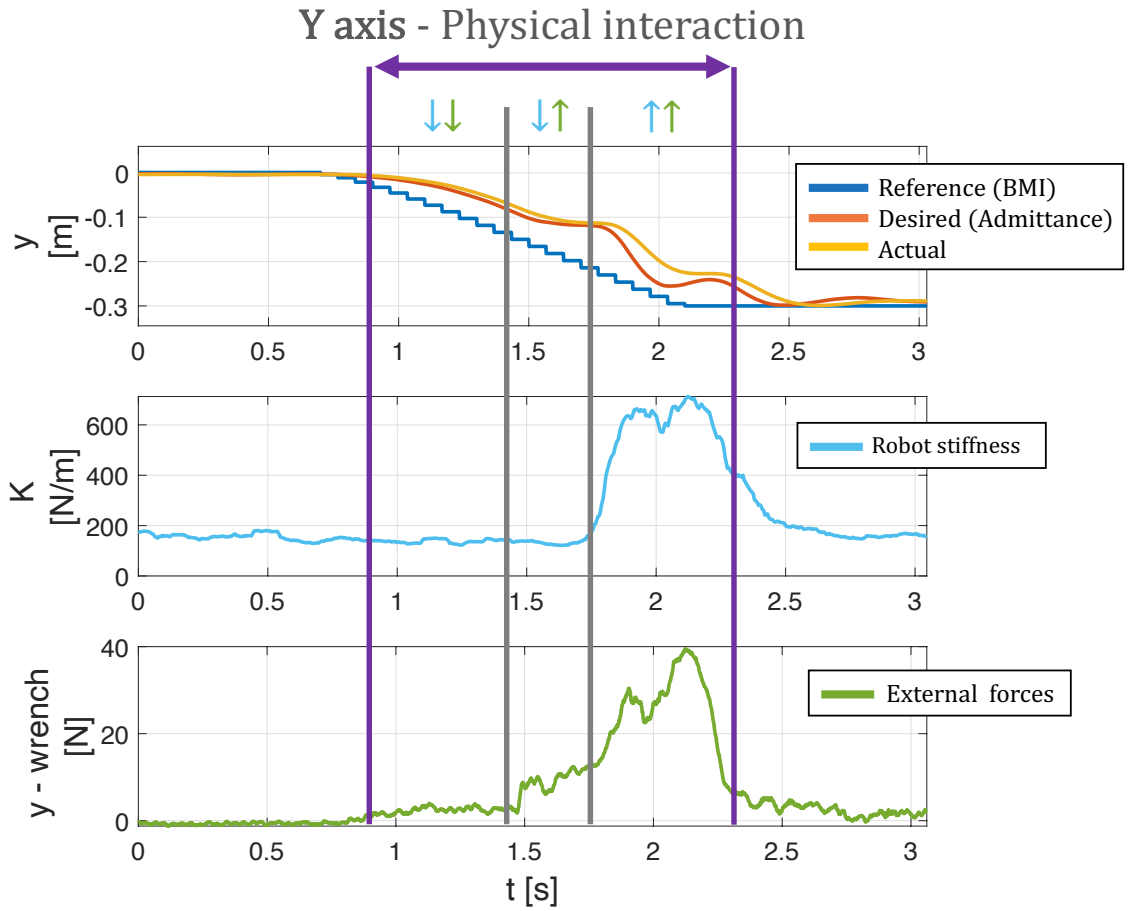


Figure 5.21: Cartesian positions tracking with adaptive admittance while an external force in Y axis interacts with the robot.

It can be seen that depending on the stiffness parameter different results on the robot y-axis position are achieved:

1. With low stiffness, a minimum physical interaction does not allow the robot to follow the desired position

2. Incrementing the force that the object exerts, the position error becomes higher

3. When the stiffness ($\boldsymbol{K}$) increments, the robot moves with higher force to follow the trajectory, that's why the value of wrench also increases

With this last implementation, it can be concluded that the hybrid control interface proposed works as expected. It is able to track desired Cartesian positions determined by body movements while adapting robot's rigidity in real-time. So, in case of detecting external forces, the robot does not continue following the trajectory if the force exerted by the operator is not high enough.

6 | Economic study and Environmental footprint

Once a project is realized, it is very important to study the effect that it has on the environment and the society. This section is in charge of first, studying both impacts and second, to determine the project budget.

6.1 Environmental impact

The robotic arm used is designed for safety, efficiency and control in real-world environments, it is: ultra-lightweight (8.2 kg), portable, power-efficient (36 W) and has the best payload-to-weight ratio. The Kinova Gen3 is made basically of carbon fibre shell and aluminium [23].

On one hand, aluminium is 100% recyclable. The process is simple and its properties aren't lost, although, it needs to be taken into account that this process, that can be done as much as wanted, requires big quantities of energy that implies in a negative impact to the ambient.

On the other hand, although the use of carbon fibre shell is expensive it is worth due to its durability. It is an environmentally friendly element: it is a very strong material that is also very lightweight and recyclable. In this process it loses some % of properties and same as with the aluminium, it wastes energy.

In terms of sensors impact:

- **Xsens MVN** inertial motion capture uses a group of straps made by a non-latex composite material and a lycra suit, that nowadays are very difficult to be recycled and also a set of motion trackers made with PC-ABS, a recyclable thermoplastic [24].

- **Myo armband** is made for a lengthy piece of flexible PCB material, which recyclability is challenging (since it contains several different chemicals, metals and glass fibres) but possible [25].

All electronic components included in each equipment count with an optimal process to their reusability and recycling, once their life cycle ends.

6.2 Socioeconomic impact

In the near future, we see an increasing amount of *cobots* helping people in their day-to-day life. These are not only highly specialized machines but rather robots that work everyday with better sensors and optimized code enabling them to accomplish more tasks than before. Such robots can empower humanity by taking on a share of the workload in fields such as healthcare or logistics, in which robots previously played little or no role.

This project tries to find solutions to society actual problems, in particular, it proposes an assistive robot to help PRM (whether disabled, elderly or otherwise), to be more autonomous, so to have a happier life and feeling independent.

This thesis shows machines ability to make people live better and also its simplicity. So, taking into account its minimum negative aspects and its ability to improve the human-robot interaction, it is considered that it has a positive overall impact on the society and environment.

6.3 Project budget

This project has been performed in Institut de Robòtica i Informàtica Industrial (IRI) during 11 months (December 2019 to October 2020), considering 4 months of teleworking due to COVID-19 situation. It has been financed with an INIREC grant of 9 months, 2 months of exceptional prorogue not covered.

Next, there is shown the price of each equipment used during this project. It has been applied a depreciation of 20 years for Kinova and 10 years for the rest.

Hardware	**Cost**
Kinova Gen3	35.000 €
Kinova computer (Linux)	600 €
Xsens MVN	40.000 €
Xsens computer (Windows)	1.200 €
Myo armband	200 €
Work computer (Mac OS)	1.450 €

Table 6.1: Hardware prices.

Apart from the hardware, the budget needs to consider the expenses regarding

MATLAB usage as software. Its annual license for individual academic use is 250€/year.

During these 11 months, it has been worked a median of 15 hours per week in the laboratory plus a median of 6 hours at home, which suppose a total of approximately 924 hours. Moreover, it has been estimated that the principal investigators, laboratory technician and research professor have dedicated 200, 100 and 10 hours, respectively.

The following table describes every project expense. Its total cost is 15.783,62 €.

Personnel		**Units**	**Unit cost**	**Total cost**
	Student	924 h	6,56 €/h	6.061,44 €
	Principal investigators	100 h	29,06 €/h	2.906,00 €
	Laboratory technician	50 h	12,25 €/h	612,50 €
	Research professor	10 h	38,75 €/h	387,50 €
Depreciation				
	Kinova Gen3	11 months	145,83 €/month	1.604,13 €
	Kinova computer	11 months	5,00 €/month	55,00 €
	Xsense MVN	11 months	333,33 €/month	3.666,67 €
	Xsense computer	11 months	10,00 €/month	110,00 €
	MYO Armband	11 months	1,67 €/month	18,37 €
	Work computer	11 months	12,08 €/month	132,88 €
Software				
	MATLAB license	11 months	20,83 €/month	229,13 €
			TOTAL	**15.783,62€**

Table 6.2: Project budget costs.

7 | Conclusions

In this project, a hybrid control interface for assistive robots has been designed and then, tested in the robotic arm: KINOVA Gen3 with 7 DoF. It combines movements and muscle activity of the upper body to control the position and the robot's rigidity, respectively.

First, a proof of concept in one robot joint has been implemented to determine the feasibility of the proposed novel approach. Next, the same performance has been transported from the joint space to the Cartesian space, to regulate better the interaction between the robot and the environment. Therefore, a Cartesian position control has been designed, simulated in MATLAB and verified experimentally in the KINOVA Gen3.

Second, an admittance control strategy has been implemented, so the robot's rigidity can be modulated using signals of muscle activity from the user. The experimental results show that this adaptive admittance strategy has potential to be very useful when interacting with the environment and performing activities of daily living with the robot.

Moreover, this project has presented a body-machine interface able to map head and shoulders movements to end-effector positions. Its functionality has been tested in a virtual environment to perform a reaching task. It is verified that the users with remaining upper body movements will be able to provide consistent position commands to the control system.

To sum up, the final control system presented in this thesis combines a Cartesian position control in real-time, with an adaptive admittance control and a body-machine interface used to give reference positions to the control. This combination allows that the system can be used to perform activities of daily living with the robot, due the user can decide the robot's position and also, modulate its impedance in real-time. So, when physical interactions appear during a trajectory the user can control the robot to behave with no rigidity and stop. Moreover, this novel approach is a good option for people with severe physical disabilities because, apart of allowing them to achieve greater independence in their daily tasks, it encourages the continued use of their motor functions: the remaining body movements for position control and muscle activity for admittance control.

Bibliography

[1] B. J. Driessen, H. G. Evers, and J. A. van Woerden. "MANUS–a wheelchair-mounted rehabilitation robot". In: *Proceedings of the Institution of Mechanical Engineers. Part H, Journal of Engineering in Medicine*. Vol. 215. 3. 2001, pp. 285–290. DOI: 10.1243/0954411011535876.

[2] V. Maheu et al. "Evaluation of the JACO robotic arm: clinico-economic study for powered wheelchair users with upper-extremity disabilities". In: *2011 IEEE International Conference on Rehabilitation Robotics*. 2011, p. 5975397. DOI: 10.1109/ICORR.2011.5975397.

[3] Amy A. Blank, Allison M. Okamura, and Louis L. Whitcomb. "Task-dependent impedance and implications for upper-limb prosthesis control". In: *The International Journal of Robotics Research* 33.6 (2014), pp. 827–846. DOI: 10.1177/0278364913517728.

[4] K. Kronander and A. Billard. "Learning Compliant Manipulation through Kinesthetic and Tactile Human-Robot Interaction". In: *IEEE Transactions on Haptics* 7.3 (2014), pp. 367–380. DOI: 10.1109/TOH.2013.54.

[5] T. Tsuj et al. "Bio-mimetic impedance control of an EMG-controlled prosthetic hand". In: *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113)*. Vol. 1. 2000, pp. 377–382. DOI: 10.1109/IROS.2000.894634.

[6] S. Rao, R. Carloni, and S. Stramigioli. "Stiffness and position control of a prosthetic wrist by means of an EMG interface". In: *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*. 2010, pp. 495–498. DOI: 10.1109/IEMBS.2010.5627153.

[7] N. Hogan. "Adaptive control of mechanical impedance by coactivation of antagonist muscles". In: *IEEE Transactions on Automatic Control* 29.8 (Aug. 1984), pp. 681–690. DOI: 10.1109/TAC.1984.1103644.

[8] A. Ajoudani et al. "TeleImpedance: Exploring the role of common-mode and configuration-dependant stiffness". In: *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*. Nov. 2012, pp. 363–369. DOI: 10.1109/HUMANOIDS.2012.6651545.

[9] J. Vogel, C. Castellini, and P. van der Smagt. "EMG-based teleoperation and manipulation with the DLR LWR-III". In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2011, pp. 672–678. DOI: 10.1109/IROS.2011.6094739.

[10] M. Ison and P. Artemiadis. "Multi-directional impedance control with electromyography for compliant human-robot interaction". In: *2015 IEEE International Conference on Rehabilitation Robotics (ICORR)*. Aug. 2015, pp. 416–421. DOI: 10.1109/ICORR.2015.7281235.

[11] S. Jain et al. "Assistive robotic manipulation through shared autonomy and a Body-Machine Interface". In: *2015 IEEE International Conference on Rehabilitation Robotics (ICORR)*. Aug. 2015, pp. 526–531. DOI: 10.1109/ICORR.2015.7281253.

[12] Ranganathan Rajiv et al. "Age-dependent differences in learning to control a robot arm using a body-machine interface". In: *Scientific Reports (Nature Publisher Group)* 9.1 (Dec. 2019).

[13] Mei-Hua Lee et al. "Body-machine interface for control of a screen cursor for a child with congenital absence of upper and lower limbs: a case report". In: *Journal of NeuroEngineering and Rehabilitation* 13.34 (2016). DOI: 10.1186/s12984-016-0139-4.

[14] Farnaz Abdollahi et al. "Body-Machine Interface Enables People With Cervical Spinal Cord Injury to Control Devices With Available Body Movements: Proof of Concept". In: *Journal of NeuroEngineering and Rehabilitation* 31.5 (2017), pp. 487–493. DOI: 10.1177/1545968317693111.

[15] Camilla Pierella et al. "Remapping residual coordination for controlling assistive devices and recovering motor functions". In: *Neuropsychologia* 79.34 (2015), pp. 364–376. DOI: https://doi.org/10.1016/j.neuropsychologia.2015.08.024.

[16] Jenifer Miehlbradt et al. "Data-driven body–machine interface for the accurate control of drones". In: *Proceedings of the National Academy of Sciences* 115.31 (2018), pp. 7913–7918. DOI: 10.1073/pnas.1718648115.

[17] E. B. Thorp et al. "Upper Body-Based Power Wheelchair Control Interface for Individuals With Tetraplegia". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 24.2 (2016), pp. 249–260. DOI: 10.1109/TNSRE.2015.2439240.

[18] B. Siciliano et al. *Robotics: Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing. Springer London, 2008. ISBN: 9781846286421.

[19] John J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[20] A. Colomé and C. Torras. "Closed-Loop Inverse Kinematics for Redundant Robots: Comparative Assessment and Two Enhancements". In: *IEEE/ASME Transactions on Mechatronics* 20.2 (2015), pp. 944–955. DOI: 10.1109/TMECH.2014.2326304.

[21] Inc. The MathWorks. *Robotic System Toolbox*. Natick, Massachusetts, United State, 2019. URL: https://www.mathworks.com/help/robotics/.

[22] Justin Carpentier, Florian Valenza, Nicolas Mansard, et al. *Pinocchio: fast forward and inverse dynamics for poly-articulated systems*. https://stack-of-tasks.github.io/pinocchio. 2015–2019.

[23] KINOVA. *Kinova Gen3 - User Guide*. URL: `https://www.kinovarobotics.com/en/resources/gen3-technical-resources`.

[24] XSENS. *Xsens MVN inertial motion capture - User Guide*. URL: `https://www.xsens.com/hubfs/Downloads/usermanual/MVN_User_Manual.pdf`.

[25] EDN. *Myo armband*. URL: `https://www.edn.com/myo-armband-wearables-design-focuses-on-packaging`.